Expert Systems with Applications 42 (2015) 7905–7916

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

CloudID: Trustworthy cloud-based and cross-enterprise biometric identification

Mohammad Haghighat [a],*, Saman Zonouz [b], Mohamed Abdel-Mottaleb [a]

[a] *Department of Electrical and Computer Engineering, University of Miami, Coral Gables, FL 33146, USA*
[b] *Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854, USA*

ARTICLE INFO

Article history:
Available online 24 June 2015

Keywords:
Biometric identification
Cloud security
Encrypted biometrics
Face recognition
Search over encrypted data

ABSTRACT

In biometric identification systems, the biometric database is typically stored in a trusted server, which is also responsible for performing the identification process. However, a standalone server may not be able to provide enough storage and processing power for large databases. Nowadays, cloud computing and storage solutions have provided users and enterprises with various capabilities to store and process their data in third-party data centers. However, maintenance of the confidentiality and integrity of sensitive data requires trustworthy solutions for storage and processing of data with proven zero information leakage. In this paper, we present CloudID, a privacy-preserving cloud-based and cross-enterprise biometric identification solution. It links the confidential information of the users to their biometrics and stores it in an encrypted fashion. Making use of a searchable encryption technique, biometric identification is performed in encrypted domain to make sure that the cloud provider or potential attackers do not gain access to any sensitive data or even the contents of the individual queries. In order to create encrypted search queries, we propose a k-d tree structure in the core of the searchable encryption. This helps not only in handling the biometrics variations in encrypted domain, but also in improving the overall performance of the system. Our proposed approach is the first cloud-based biometric identification system with a proven zero data disclosure possibility. It allows different enterprises to perform biometric identification on a single database without revealing any sensitive information. Our experimental results show that CloudID performs the identification of clients with high accuracy and minimal overhead and proven zero data disclosure.

1. Introduction

Substitution of biometrics for passwords in authentication and identification systems received attention in security systems (Jain, Ross, & Pankanti, 2006). Biometric identifiers are distinctive and measurable characteristics used to recognize individuals (Jain, Hong, & Pankanti, 2000a). Some of the well-known biometrics used for human identification are fingerprints, face, iris, voice and DNA. Some of the advantages of biometrics over passwords include their higher level of security, mobility, difficulty to forge, and user friendliness. According to a new study published by Javelin Research (Javelin Strategy & Research, 2014), smartphone users prefer using biometrics as an alternative for passwords, which brings more security to new technologies such as *Apple Pay*. In spite of all these advantages, there are a few challenges that biometric systems face.

One of the major challenges of biometric systems is the variability in the characteristics of the biometrics for each individual. Human face, as an example biometric trait, is a complex object with features that change over time. Facial features change due to changes in illumination, head pose, facial expressions, cosmetics, aging, and occlusions because of beard or glasses. However, we humans have an ability to recognize faces and identify persons at a glance. This natural ability does not exist in machines; therefore, we design intelligent and expert systems that can simulate the recognition artificially (Khashman, 2008). These intelligent systems are trained using the biometric information of the subjects who are enrolled in the system. The identification is performed through a comparison between the biometric information of the query subject and the enrolled subjects. Therefore, these systems need to store biometric information of all enrolled subjects in databases to be utilized at the time of query.

As the number of subjects increases, the system requires more storage capacity and more processing power. On the other hand, these databases need to be accessible by all enterprises that make use of biometric identification. The need to be accessed by multiple

* Corresponding author.
E-mail addresses: haghighat@umiami.edu (M. Haghighat), saman.zonouz@rutgers.edu (S. Zonouz), mottaleb@miami.edu (M. Abdel-Mottaleb).

http://dx.doi.org/10.1016/j.eswa.2015.06.025
0957-4174/© 2015 Elsevier Ltd. All rights reserved.

enterprises and to have a high processing power motivate the use of a cloud-based system to store and process the data. The work presented in this paper is a key step towards a cloud-based unified storage system for personal records. The idea is to create an encrypted database of personal records of individuals, *e.g.*, name, date of birth, educational information, banking or credit history, medical records, criminal records, insurance, etc., as a unified and privacy-preserving cloud-based database. Biometric information of individuals are used as a key attribute for this database.

The growing popularity of cloud-based systems has increased the importance of addressing the serious issue of the security of the data stored in the cloud (Fernandes, Soares, Gomes, Freire, & Inácio, 2014; Kandukuri, Paturi, & Rakshit, 2009; Padilha & Pedone, 2015; Ren, Wang, & Wang, 2012; Takabi, Joshi, & Ahn, 2010). In case of using biometrics to have access to records stored on the cloud, there is the risk of identity theft, because biometric data of the enrolled subjects can be stolen and misused against their will. The biometric data is unique and irrevocable, and unlike passwords users cannot change their biometrics. Consequently, the system must guarantee the preservation of the users' privacy, and therefore, the biometric database has to be encrypted.

Since the encrypted biometric database is stored in a public cloud, the identification process should be done with minimum amount of information leakage. That is, the comparisons need to be performed without the decryption of the data to prevent eavesdroppers from any access. However, the variations in the biometrics of each subject bring about a serious problem in the encrypted domain. Small changes in the data (plaintext) result in big differences in the ciphertext (encrypted plaintext). This difference can mislead the recognition process. Consequently, it is not feasible to just simply add an encryption scheme to a biometric identification system in order to secure the data and expect to obtain the same results that are obtained without the encryption.

Contributions: In this paper, we present a privacy-preserving cloud-based identification system (CloudID), which allows users to securely store their confidential information in untrusted public clouds. It gives them the opportunity of having effective and secure storage along with the computational power of the cloud infrastructures as well as controlled and flexible data access by multiple agents. Our proposed approach is the first cloud-based biometric identification system with a proven zero data disclosure possibility. CloudID performs the identification process in the encrypted domain without decrypting the data. This prevents the cloud provider or potential attackers from gaining access to any sensitive data or even the contents of the individual queries.

Unlike other privacy-preserving biometric identification methods, our approach does not apply a distance-based matching. However, using the query sample, it creates an encrypted conjunctive range query, which is applied on the encrypted gallery samples stored in the cloud. In this scenario, the only revealed piece of information is the binary matching result, *i.e.*, *match* or *not match*. This makes CloudID secure against *center search attack* (Pagnin, Dimitrakakis, Abidin, & Mitrokotsa, 2014) in which the attacker can recover the biometric template even if it is stored encrypted. In order to improve the performance of the biometric-based identification in the encrypted domain, we propose a k-d tree structure to create encrypted search queries. Applying this structure in the core of a searchable encryption technique helps the system not only to quantize the biometric features but also to handle the variations in the biometric data. Moreover, our algorithm is not limited to any specific type of biometric data and it can work with any biometric trait and any feature extraction method.

A working prototype of the CloudID framework is implemented and evaluated using a public biometric database. Our experimental results show the feasibility of CloudID for accurate biometric identification with no confidential data disclosure possibility, which enables building trustworthy storage systems for sensitive records.

The rest of the paper is organized as follows. Section 2 presents related work from the literature. Section 3 provides a brief overview of the system whose complete design is described in Sections 4 and 5. The implementation details and overall performance of the system are presented in Section 6. Finally, Section 7 concludes the paper.

2. Related work

In the literature, there are two sets of studies considering the privacy-preserving biometrics identification. The first set of studies (Barni et al., 2010; Blanton & Gasti, 2011; Erkin et al., 2009; Huang, Malka, Evans, & Katz, 2011; Osadchy, Pinkas, Jarrous, & Moskovich, 2010, 2013; Sadeghi, Schneider, & Wehrenberg, 2009) only achieve privacy-preserving at the time of executing the query, protecting the confidentiality of both server and client. In these approaches, the server, in which the biometric database is stored, is considered to be trusted, and the biometric database is stored unencrypted. This allows the server to have access to the contents of the biometric database. However, these approaches cannot be used in case of untrusted servers such as clouds. In the second set of studies, the server is not trusted and the biometric database is encrypted (Bringer, Chabanne, & Kindarji, 2009, 2011, 2013a, 2013b, 2014). However, these algorithms have some limitations and suffer from information leakage. In this section, we briefly describe these methods and compare them with our proposed approach.

To the best of our knowledge, Erkin et al. (2009) considered the problem of privacy preserving biometric identification for the first time. They proposed a privacy-preserving face recognition system based on the well-known eigenface approach introduced by Turk and Pentland (1991a, 1991b). They employed Pailliers cryptosystem (Paillier, 1999), as an additive homomorphic encryption and calculated the Euclidean distance between face image feature vector from client and server's face image database. The matching algorithm is performed between the client and the server without revealing the client's biometric information or the result of the query to the server. At the same time, the client cannot learn from the database stored in the server. Later, Barni et al. (2010) proposed a similar algorithm for a fingerprint recognition system, *FingerCodes* (Jain, Prabhakar, Hong, & Pankanti, 2000b). Both of these protocols (Barni et al., 2010; Erkin et al., 2009) rely on homomorphic encryption and do not try to find the specific match but the group of the nearest matches within some threshold.

Sadeghi et al. (2009) improved the efficiency of Barni's algorithm by applying a hybrid approach where *garbled circuits* were used in conjunction with homomorphic encryption to find the minimum distance. Huang et al. (2011) combined the algorithms proposed in Sadeghi et al. (2009) and Erkin et al. (2009) to further improve the computational and bandwidth efficiency of the system.

The main idea in Erkin et al. (2009), Sadeghi et al. (2009) and Huang et al. (2011) is to find the nearest match for a query in the biometrics database based upon the Euclidean distance. In these references, each query is encrypted using the public key published by the client and sent to the server. The server also encrypts each biometric data in the database using an additive homomorphic encryption using the same public key. Then, the Euclidean distances between the query and each gallery in the database are calculated in the encrypted domain, $d_1, d_2, \ldots, d_n$. In Sadeghi et al. (2009) and Huang et al. (2011), this information is fed into a garbled circuit, which finds the closest match by calculating $i^* = argmin_i(d_1, d_2, \ldots, d_n)$.

Blanton and Gasti (2011) also proposed a secure protocol for iris codes based on additive homomorphic encryption and garbled circuits. The protocol uses Hamming distance to measure the similarity between iris codes. They also applied their technique on *FingerCode* calculating the Euclidean distances for fingerprint recognition.

Osadchy et al. (2010, 2013) designed a privacy-preserving face identification system called SCiFI. The implementation of this method is based on additive homomorphic encryption and oblivious transfer. SCiFI represents the facial images by binary feature vectors and uses Hamming distances to measure the image similarity.

In the above-mentioned scenarios (Barni et al., 2010; Blanton & Gasti, 2011; Erkin et al., 2009; Huang et al., 2011; Osadchy et al., 2010, 2013; Sadeghi et al., 2009), the server cannot learn neither the client's biometric information nor the query result. On the other hand, the client does not get any information about the biometric database, which is saved unencrypted in the trusted server. However, these approaches cannot be used in case of untrusted servers like public clouds. In CloudID, our assumptions are quite different from these protocols. In our approach, the biometric database is encrypted and outsourced to an untrusted server (cloud), which does not have the key to decrypt the data. Therefore, not only the query process but also the biometric database stored in the server is secure. This makes our approach applicable for secure storage and processing of biometric data in public clouds.

Another group of researchers also considered the problem of privacy-preserving biometric identification (Bringer et al., 2009, Bringer, Chabanne, & Kindarji, 2011, 2013a, Bringer, Chabanne, & Patey, 2013b, 2014). These studies are much similar to our approach since the biometric database is encrypted and the identification process is applied in the encrypted domain without decrypting the data. Bringer et al. (2009, 2011) introduced an error-tolerant searchable encryption system to cope with the intra-class variations of the biometric data. This method is designed for a special type of biometric data, *IrisCode*, in which biometric templates are represented by binary vectors (Daugman, 1993). With the use of binary representation, comparisons are performed by Hamming distance. A locality sensitive hashing function is applied to narrow down the database search to a few candidates. The locality sensitive hashing function produces identical or very similar hash results for representations that are close to each other. This actually categorizes data in neighborhoods that have a few members. Given a query, it is assumed that the answer is among a set that has the same hash as the query. From this set, the nearest neighbor to the query is considered as the definite answer. Being computationally expensive, they improved the performance of this algorithm using Oblivious RAM (Bringer, Chabanne, & Patey, 2013a), and Oblivious Transfer (Bringer et al., 2013b). In Bringer et al. (2014), they generalized the method proposed in Bringer et al. (2013b) to compute several distance metrics, such as Hamming distance, Euclidean distance, and Mahalanobis distance.

The algorithms proposed in Bringer et al. (2009, 2011, 2013a, 2013b, 2014) do not guarantee zero data disclosure and suffer from information leakage. The mathematical framework to analyze the information leakage is provided in Pagnin et al. (2014). Pagnin et al. (2014) employ a *center search attack* and prove that when the matching process is performed using a specific family of distances (such as Hamming and Euclidean distances), then information about the reference template is leaked, and the attacker can recover the biometric template even if it is stored encrypted. On the other hand, using the locality sensitive hashing function, the (potentially malicious) cloud provider is able to cluster the data into groups based on the pattern for searching/processing the data for every received query. Therefore, through observations of the data search procedure, the malicious provider may eventually be able to sort the data records, which could lead to potential data confidentiality breach. To address these vulnerabilities, our proposed method (CloudID) does not calculate any distances to match the biometric information. CloudID applies a conjunctive query over encrypted biometrics, and the only information that is revealed is the binary matching result, *i.e.*, *match* or *not match*.

3. System overview and access control

CloudID is the first practical system that provides privacy-preserving capability to a biometric-based system for cloud-side query processing.[1] Fig. 1 illustrates the different components of CloudID and their logical interconnections. The database of personal records including their biometric data is stored encrypted in the cloud to prevent attackers from gaining unauthorized access. Biometric identification is applied in the encrypted domain to ensure that the cloud provider or potential attackers do not gain access to any sensitive data. On the other hand, the system must ensure that the cloud provider cannot learn the contents of individual queries either. When a query is performed, the provider should learn nothing but the value of a conjunctive predicate. We modify a search-aware encryption technique (Song, Wagner, & Perrig, 2000; Boneh & Waters, 2007) to make it applicable for biometric data.

The proposed system makes use of a *multiple-encryption* technique, *i.e.*, encrypting an already encrypted message. The two stages of the multiple-encryption are called *inner-level encryption* and *outer-level superencryption*, in order of occurrence. During the inner-level encryption, our system encrypts all personal information of subject i, *e.g.*, his/her medical records (M_i), using the public key (PK_e) provided by each enterprise, *e.g.*, health care organization. εM_i denotes the encrypted data obtained by the inner-level encryption.

$$\varepsilon M_i \xleftarrow{\text{inner-level-enc.}} Enc(PK_e, M_i). \tag{1}$$

Then, during the outer-level superencryption, CloudID links the data encrypted in the inner-level (εM_i) to the subject's biometric record (B_i). That is, CloudID generates a public key based on the biometric record of the subject (PK_{B_i}) and encrypts all his/her information before storing it in the cloud. C_i denotes the ciphertext, corresponding to subject i, stored in the cloud.

$$C_i \xleftarrow{outer-level-superenc.} SuperEnc(PK_{B_i}, \varepsilon M_i). \tag{2}$$

The link to the biometric data enables the system to identify the subject and retrieve the corresponding data from the database.

Although CloudID can be extended to various biometric identification systems, we use facial data as an example in this paper. As shown in Fig. 1, at the time of a query, a camera captures the image of the client and sends it to the biometric data acquisition system to localize the region of interest in the image. In case of facial data, a face detector localizes the face, then the face region is fed into the feature extraction stage. The number of the extracted features is usually high, which reduces the performance of the system. Therefore, a dimensionality reduction method is used to reduce the length of the feature vector. The feature vectors consist of real numbers; however, encryption algorithms are applied to discrete values. We propose a k-d tree structure to quantize the features and create encrypted search queries at the same time.

Since the biometric features of a subject change over time, the system derives range boundaries from the feature vector of the query to be compared with the feature vectors stored in the database. A conjunctive predicate, $P()$, is created for the comparison query. Eq. (3) shows the conjunctive proposition for a biometric

[1] A preliminary version of this work appeared in a conference paper (Haghighat, Zonouz, & Abdel-Mottaleb, 2013).

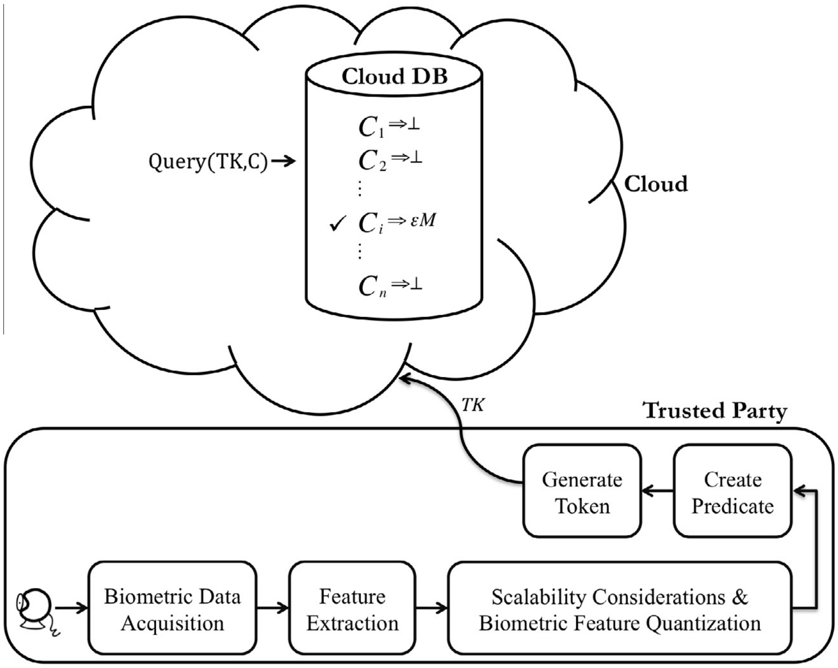

Fig. 1. Framework of the CloudID in query processing.

feature vector $B = [b_1, b_2, \ldots, B_n]^T$. The query is answered by records whose feature vectors fall in between these boundaries.

$$P(B) = \begin{bmatrix} l_1 < b_1 < u_1 \\ l_2 < b_2 < u_2 \\ \vdots \\ l_n < b_n < u_n \end{bmatrix} \tag{3}$$

In order to achieve maximum protection and minimum amount of information leakage, rather than verifying the individual comparison propositions separately, we verify the whole conjunctive proposition. For example, in the above predicate, where the length of the feature vector is n, $2n$ propositions must be verified conjunctively $(l_1 < b_1 \wedge b_1 < u_1 \wedge l_2 < b_2 \wedge b_2 < u_2 \wedge \ldots \wedge l_n < b_n \wedge b_n < u_n)$. From the cloud's point of view, such a query only reveals the Boolean value of the conjunctive predicate. That is, if all the propositions are true, then the query result is εM; however, if any of the propositions is false, then the query will return NULL output ($\perp$), and the cloud provider will not know which comparison(s) is/are not true. For example, if a conjunctive query, *e.g.*, $P_1 \bigwedge P_2$, is false, the cloud provider should not be able to tell whether P_1 or P_2 or both were false for the conjunction to be false. All the steps illustrated in Fig. 1 are described in detail in Sections 4 and 5.

We assume that all enterprises have access to the public cloud on which all the users' encrypted confidential information is stored. For example, once Alice grants access to a specific enterprise, the cloud gives the enterprise permission to make queries over her encrypted data. In other words, each enterprise has the right to make queries over the data of only a subset of the users who granted access to the enterprise. Alice can also revoke the access by asking the cloud to remove her data from the list of records accessible by the enterprise. The access control for the enterprises in the cloud is out of the scope of this paper. Instead, we focus on the privacy of the encrypted information and provide a search over encrypted data scheme that can securely find the information of the users using their biometric data.

4. Offline system setup

In this section, we describe the offline setup of the privacy-preserving cloud-based identification solution. As mentioned above, we use face images as the biometric identifiers.

The biometric identification in CloudID consists of two phases: training and query. During the training phase, the system stores the feature vectors of the individuals in a biometric database, while in the query phase, it identifies the closest match to the query from the database. The number and type of features used have a direct effect on the performance and accuracy of the identification.

After detecting the face, features are extracted from the face region, then the dimension of the feature space is reduced and the features are quantized in preparation for encryption. The components of the CloudID are described in the following subsections.

4.1. Biometric data acquisition

The first stage of gathering face information for a subject is to capture a photo of the subject and localize the face area using a face detection algorithm. CloudID employs the Viola–Jones method (Viola & Jones, 2004) for face detection. Fig. 2 shows a few examples of face detection results, obtained by this algorithm, on images from the FERET database (Phillips, Moon, Rizvi, & Rauss, 2000).

4.2. Feature extraction

CloudID uses Gabor filters to extract features from the detected face region. The most important advantage of Gabor filters is their invariance to rotation, scale, and translation. Furthermore, they are robust against photometric disturbances, such as illumination changes and image noise (Kamarainen, Kyrki, & Kalviainen, 2006; Liu & Wechsler, 2002; Meshgini, Aghagolzadeh, & Seyedarabi, 2013; Shen, Bai, & Fairhurst, 2007).

The Gabor filter-based features are directly extracted from the gray-level images. In the spatial domain, a two-dimensional Gabor filter is a Gaussian kernel function modulated by a complex sinusoidal plane wave, defined as:

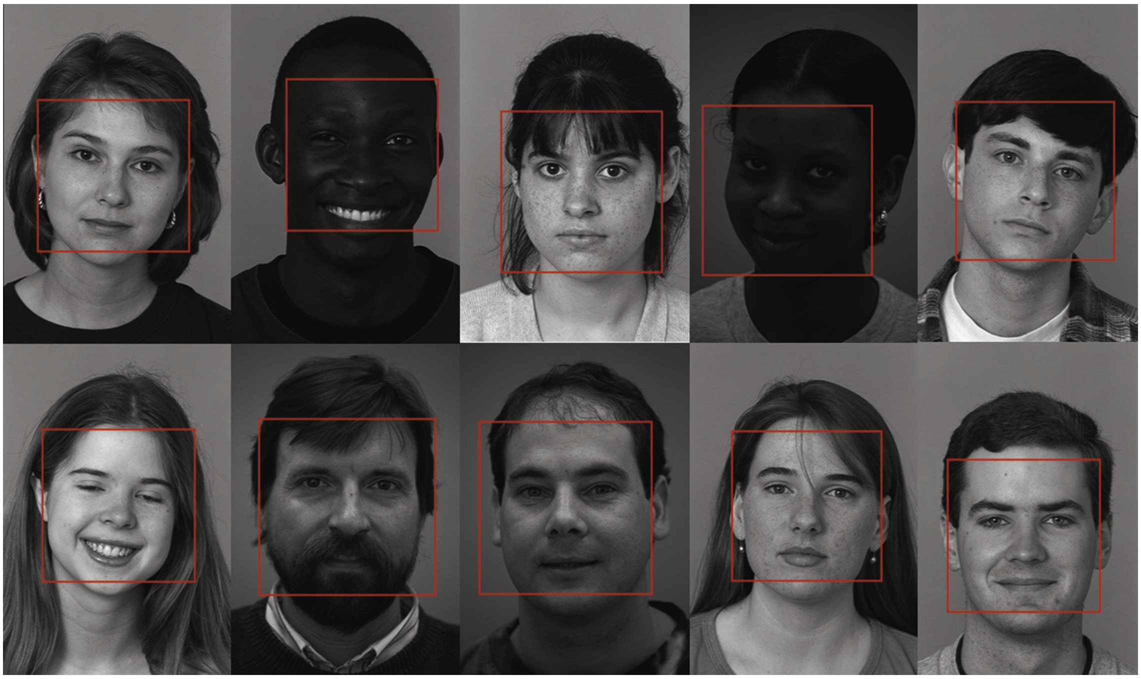

Fig. 2. Face detection using Viola and Jones method (Viola & Jones, 2004).

$$G(x,y)=\frac{f^2}{\pi\gamma\eta}exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)exp(j2\pi fx'+\phi)$$
$$x'=xcos\theta+ysin\theta$$
$$y'=-xsin\theta+ycos\theta \tag{4}$$

where f is the frequency of the sinusoid, θ represents the orientation of the normal to the parallel stripes of a Gabor function, ϕ is the phase offset, σ is the standard deviation of the Gaussian envelope and γ is the spatial aspect ratio which specifies the ellipticity of the support of the Gabor function.

CloudID employs forty Gabor filters in five scales and eight orientations as shown in Fig. 3. The size of the face images used in our experiments is 120×120 pixels. Using forty Gabor filters, the dimension of the feature vector is $120\times120\times40=576,000$. Since the adjacent pixels in an image are usually highly correlated, we can reduce this information redundancy by downsampling the feature images resulting from Gabor filters (Liu & Wechsler, 2002; Shen et al., 2007). The feature images are downsampled by a factor of four, which means that the feature vector will have a size of $576,000/(4\times4)=36,000$. These vectors are then normalized to zero mean and unit variance. In addition to downsampling, we need to use dimensionality reduction methods to further reduce the size of the feature vectors.

4.3. Scalability considerations and feature quantization

In biometrics, the number of extracted features is usually high, which increases the computational complexity and decreases the performance of the system because of the curse of dimensionality. In order to address these issues, dimensionality reduction methods

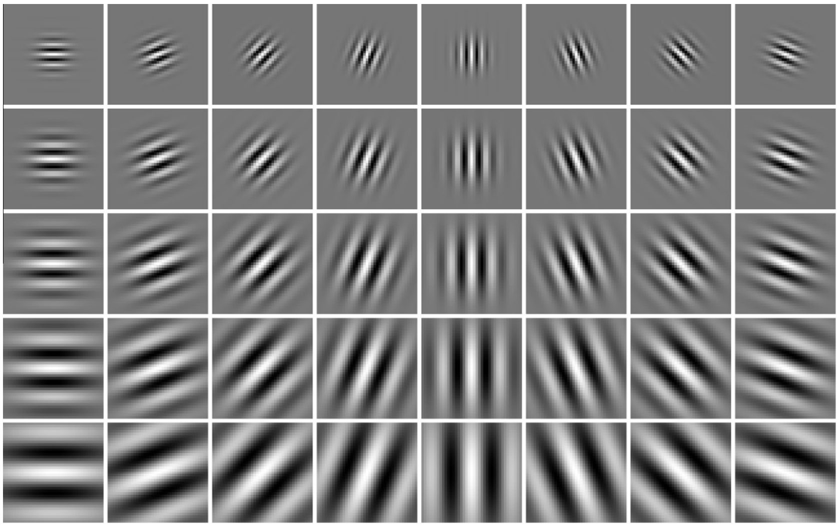

Fig. 3. Gabor wavelets in five scales and eight orientations.

are applied to reduce the length of the feature vectors and their redundancies (Haghighat & Namjoo, 2011).

Classical dimensionality reduction techniques, such as principal component analysis (PCA) and linear discriminant analysis (LDA), have been widely utilized in face recognition techniques since Turks pioneering work (Turk & Pentland, 1991a). However, each of these well known methods has its own shortcomings.

The Eigenface method, which uses PCA for dimensionality reduction, results in projection directions that maximize the total scatter across all classes, *i.e.*, across all images of all faces. During this projection which maximizes the total scatter, PCA preserves unwanted variations due to lighting and facial expressions (Belhumeur, Hespanha, & Kriegman, 1997). As stated in Moses, Adini, and Ullman (1994), "the variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to change in face identity".

The Fisherface method, on the other hand, uses LDA to find a projection matrix W which is optimized to separate different classes, *i.e.*, maximizes the ratio of between-class scatter (S_b) to within-class scatter (S_w):

$$W = \arg max \frac{|W^T S_b W|}{|W^T S_w W|}. \tag{5}$$

W can be computed from the eigenvectors of $S_w^{-1} S_b$. However, due to the curse of dimensionality, especially in case of face recognition with low number of training data, S_w is usually singular, *i.e.*, the inverse of S_w does not exist.

To overcome these shortcomings, CloudID uses generalized discriminant analysis (GDA) (Baudat & Anouar, 2000), which is a non-linear feature reduction technique. Similar to LDA, the objective of GDA is to find a projection for the features into a lower dimensional space by maximizing the ratio of between-class scatter to within-class scatter. Suppose that the space X is mapped into the space F through a non-linear mapping function $\phi : X \rightarrow F, x \rightarrow \phi(x)$. Considering C to be the number of classes and N_z to be the number of samples in class z, the S_w and S_b of the training set can be computed as follows:

$$S_w = \frac{1}{C}\sum_{z=1}^{C}\frac{1}{N_z}\sum_{k=1}^{N_z}\phi(x_{zk})\phi^t(x_{zk}) \tag{6}$$

$$S_b = \frac{1}{C}\sum_{z=1}^{C}(\mu_z - \mu)(\mu_z - \mu)^t \tag{7}$$

where μ_z is the mean of the samples that belong to class z. GDA finds the eigenvalues λ and eigenvectors v that satisfy:

$$\lambda S_w v = S_b v. \tag{8}$$

Since the eigenvectors lie in the span of $\phi(x_{11}), \dots, \phi(x_{zk}), \dots, \phi(x_{CN_z})$, there exists α_{zk} such that

$$v = \sum_{z=1}^{C}\sum_{k=1}^{N_z}\alpha_{zk}\phi(x_{zk}). \tag{9}$$

To generalize LDA to the nonlinear case, GDA considers an expression of dot product of a sample i from class p and another sample j from class q by the following kernel function

$$(k_{ij})_{pq} = \phi^t(x_{pi}).\phi(x_{qj}) = k(x_{pi}, x_{qj}) = e^{-|x_{pi}-x_{qj}|^2/r}. \tag{10}$$

Let $\mathbf{K}$ be an $M \times M$ matrix defined on the class members by $(\mathbf{K}_{pq})_{p=1,\dots,C,\ q=1,\dots,C}$, where $\mathbf{K}_{pq}$ is a matrix composed of dot products between class p and q in the feature space F:

$$\mathbf{K}_{pq} = (k_{ij})_{i=1,\dots,N_p,\ j=1,\dots,N_q}. \tag{11}$$

We also introduce an $M \times M$ block diagonal matrix:

$$\mathbf{U} = (\mathbf{U}_z)_{z=1,\dots,C} \tag{12}$$

where $\mathbf{U}_z$ is an $N_z \times N_z$ matrix with all its elements equal to $\frac{1}{N_z}$.

By substituting Eqs. (6), (7), and (9) into (8) and taking inner-product with vector $\phi(x_{ij})$ on both sides, the solution of (8) can be obtained by solving:

$$\lambda \mathbf{KKa} = \mathbf{KUKa}. \tag{13}$$

where $\mathbf{a}$ denotes a column vector with entries $\alpha_{zk}, z = 1, \dots, C, k = 1, \dots, N_z$. The solution of $\mathbf{a}$ is computed by finding the eigenvectors of the matrix $(\mathbf{KK})^{-1}\mathbf{KUK}$. If matrix $\mathbf{K}$ is not reversible, the eigenvector is found by first diagonalising matrix $\mathbf{K}$ (Baudat & Anouar, 2000). After finding the L most significant eigenvectors, a projection matrix is constructed as:

$$W = [\mathbf{a}_1\mathbf{a}_2 \dots \mathbf{a}_L]. \tag{14}$$

The projection of x in the L-dimensional GDA space is calculated by:

$$y = k_x W \tag{15}$$

where $k_x = [k(x, x_{11}) \dots k(x, x_{zk}) \dots k(x, x_{CN_c})]$.

Note that the number of features in LDA-based methods can be at most $C - 1$. In our experiments with the database of 200 subjects, the maximum size of the projected vectors is 199 which is a significant reduction in comparison to 36,000.

CloudID combines the feature quantization, feature selection, and classification using parallel k-d trees. The feature vectors of the training set are organized in a k-d tree by partitioning the feature space along each feature using a hyperplane. In order to quantize L features, we need a k-d tree of the depth $L + 1$, which stores up to $2^L - 1$ samples. The maximum number of quantized features is calculated using Eq. (16).

$$L_{max} = \lceil \log_2(C + 1) \rceil - 1 \tag{16}$$

In our experiments, since the number of classes, C, is 200, the maximum possible depth of the k-d tree is 8, which can quantize 7 features. In order to make use of more features, CloudID employs several k-d trees in parallel for different sets of features. Fig. 4 illustrates the process of creating a quantized feature vector using several k-d trees.

Since GDA sorts the features according to their discriminative power, CloudID selects the most discriminative features of the GDA to construct the k-d trees. Moreover, it builds the k-d tree such that the less discriminative feature is used for the root and the most discriminative feature for the leaves. The feature used in the root of a tree only defines one hyperplane and divides the space into two partitions. Consequently, the second and third levels of the tree divide the feature space into just three and five partitions. Since the features near the root of the tree only define a few quantization levels, CloudID assigns the least discriminative features for those levels and do not use them in the quantization process. On the other hand, the most discriminative features are used in deeper levels of the tree where the number of hyperplanes are high and a finer quantization is performed.

As shown in Fig. 4, using the training set, CloudID makes use of 91 features to construct 13 parallel k-d trees. Each of the first 13 features of the GDA output, *i.e.*, the most discriminative ones, is used for the leaves of a tree. On the other hand, the last 13 features (79 to 91) are used for the roots. The first three levels of each tree are then disregarded and not included in the conjunctive predicate. In other words, each tree is used to quantize 4 features and therefore by using 13 trees we are quantizing the first 52 features of the GDA output. Note that, in real systems, the number of subjects are higher, which, based on Eq. (16), increases the number of features used in each tree.

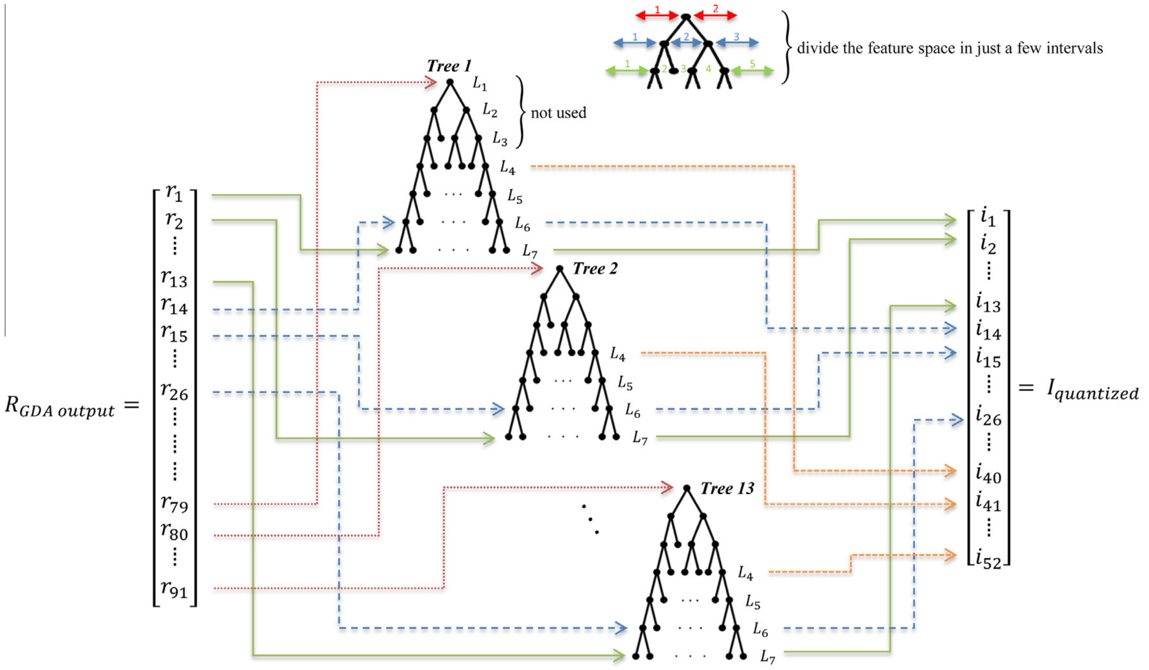

Fig. 4. Creating the quantized feature vector using 13 k-d trees of 4 significant features.

4.4. Cryptographic key generation

Generally, owners of the biometric records cannot trust cloud providers, even the well known providers, for storing their data in plain format, due both to internal security policies, and federal or state laws and requirements (University of California Corporate Compliance Policies & Procedures, 2008). Therefore, data must be encrypted before storing it on a remote server.

While the usage of regular cryptographic encryption solutions prevents unauthorized access and helps ensure the confidentiality of biometric data, it prevents services such as searching over encrypted data that is required for secure identification in the CloudID framework. As a result, the whole biometric data set must be downloaded to a local system for every query. This method is not efficient in terms of bandwidth or computational requirements on the client side because the client has to store and decrypt all received data to be able to extract those that are actually needed.

To provide trustworthy data storage and data search capabilities, CloudID makes use of search-aware encryption (Boneh & Waters, 2007; Song et al., 2000) to prevent attackers from gaining unauthorized access to the confidentially-sensitive biometric data records. Let us assume that the trusted party intends to encrypt and send a user's biometric data record, denoted by B, to the cloud for secure storage and remote biometric matching that requires query processing over encrypted data. This approach consists of four major phases, namely *Setup*, *Encryption*, *Token Generation*, and *Query*. The first two phases are parts of the offline system setup and the latter two phases are parts of the online setup and will be described in the next section.

CloudID uses a public/private key scheme to encrypt and decrypt data. In the *Setup* phase, all the cryptographic parameters, *i.e.*, public key, PK, and private key, SK, are generated:

$$PK \leftarrow (PK_1, PK_2, \dots, PK_t), \tag{17}$$

$$SK \leftarrow (SK_1, SK_2, \dots, SK_t), \tag{18}$$

where t denotes the number of all possible predicates for which the user can query. t directly depends on the number of features and the number of quantization levels for each feature.

4.5. Biometric record encryption

The *Encryption* phase performs the last offline setup step by encrypting each user's biometric data record. It is important to note that this step will be completed locally by the trusted party so that the (untrusted) cloud provider will only have access to ciphertexts, and not the encryption keys themselves. In particular, the *Encryption* step is carried by using the following assignments for $1 < j < t$

$$C_j \leftarrow \begin{cases} Encrypt(PK_j, M) & \text{if } P_j(B) = 1 \\ Encrypt(PK_j, \perp) & o/w, \end{cases} \tag{19}$$

where M is the relevant data of the subject that is encrypted using the public key, PK_j, corresponding to the predicate, $P_j()$, which is satisfied by user's biometric, B (refer to Eq. (3)). The other public keys corresponding to the *False* propositions will encrypt a NULL value ($\perp$). Once completed, this step will give us a ciphertext vector

$$C \leftarrow (C_1, C_2, \dots, C_t), \tag{20}$$

whose length is the number of possible predicates t. Note that, only one of the ciphertexts includes the data of the subject.

5. Online system behavior

After the offline system setup and storing the biometric samples of the individuals in the database, the system is ready to respond to queries and identify a user via his/her biometric information. Here, we assume that all the subjects are already registered in the database by their face information. All the relevant data, M, belonging to a subject is linked to his/her biometric, B, and both the data and the biometric are encrypted. During the query time, a subject's biometric information is captured by a camera and CloudID compares it to encrypted biometrics in the database to identify the person. The biometric matching is performed in encrypted domain so that neither an intruder nor the cloud provider can access the user information.

5.1. Face features variations

As mentioned before, the biometrics of an individual change over time. For example, facial features vary due to the changes in illumination, head pose, facial expressions, cosmetics, aging, and occlusions. Therefore, the feature vector of the query will not be exactly the same as the one saved in the database for the same person.

In the feature quantization section, we describe how the CloudID uses k-d trees to find the quantization thresholds. Some variations in the facial features may result in significant changes in the quantized data, *i.e.*, a particular feature value in the query may not fall in the same quantization interval of the corresponding feature for the same subject enrolled in the database. Therefore, the system needs to accept not only the exact quantized values, but reasonably-sized ranges in order to include accepted variations around the quantized values. These ranges are defined by lower and upper boundaries, where the features of the query fall in between these two boundaries. Note that the size of these ranges has a direct influence on the accuracy of the system. A tight range makes the system very strict not accepting large variations; and a wide range makes the system more lenient accepting large variations. Therefore, in the first case, the recognition rate is low, however in the latter case, the recognition rate increases at the cost of a higher false positive rate.

As shown in Fig. 4, k-d trees divide the feature space into several intervals. The above-mentioned ranges are made of few adjacent intervals around the query's feature values. The number of the intervals in each level, L, of the tree is $2^{L-1}+1$; *i.e.*, the features near the root of the tree divide the space into less intervals than the ones near the leaves. Therefore, the intervals near the root of the tree are wide and coarse while at deeper levels they are fine. Taking this into account, CloudID increases the number of the accepted adjacent intervals in deeper levels of the tree to make sure that we accept almost the same size ranges in all levels. In our experiments, we have empirically chosen the size of the range to be $2^{L-2}+2$ adjacent intervals.

5.2. Cryptographic token generation

To preserve the users' privacy, CloudID employs a searchable encryption for comparison queries (Boneh & Waters, 2007). The *Token Generation* phase prepares the system for processing a given query. As discussed above, for each query, CloudID creates numerical ranges defined by lower and upper boundaries. CloudID processes the comparison queries given the range boundaries applied in a conjunctive manner to create a predicate, $P()$. CloudID implements *GenToken* that uses each query predicate and the secret key to generate the corresponding *token TK* and sends it to the cloud.

$$TK \leftarrow GenToken(SK, P()). \tag{21}$$

This step is accomplished locally by the trusted party, and hence the cloud provider will only see the encrypted token. For instance, given predicate $P_i()$, the trusted party chooses the corresponding private key $SK_i \in SK$[2] to generate the token. This *token* will be used in the *Query* phase.

5.3. Biometric database query

The final step is processing the received query by the cloud provider that has access to the encrypted database records, and the single token, TK sent by the trusted party. The cloud provider uses TK to decrypt the corresponding ciphertext C_i for each individual's records. Consequently, the cloud provider will only retrieve the records if the biometric of the individual satisfies the predicate used for the token generation, and will get $\perp$ as a result for all other decryptions (Eq. (19)).

$$\begin{cases} Query(TK, C) = M & \text{if } P(B) = 1 \\ Query(TK, C) = \perp & \text{if } P(B) = 0 \end{cases} \tag{22}$$

There might be more than one biometric record in the database that satisfy the query predicate. All the retrieved records are then sent back to the trusted party and filtered after decryption.[3]

In summary, using the above-mentioned four steps, CloudID makes sure that the cloud provider can perform and accomplish the search over the encrypted data, and in the meantime, cannot access any confidential data. See Appendix A for the proof of user privacy preservation in CloudID.

6. Evaluation

6.1. Experimental setup

We evaluated the accuracy and performance of CloudID's various components through an extensive set of experiments. To that end, we used *Pegasus* high performance computing cluster at the center for computational science (CCS) at University of Miami (University of Miami, 2015). Each node in Pegasus is equipped with eight Intel® Xeon® Sandy Bridge 2.6 GHz cores and 32 GB of memory.

In order to evaluate the system, the Facial Recognition Technology (FERET) database is used in our approach (Phillips et al., 2000). Six hundred frontal face images from 200 subjects are selected, *i.e.*, three images per subject: a frontal image with neutral expression, a frontal image with an alternative expression, and a frontal image with different illumination. In FERET database, these images are letter coded as *ba*, *bj*, and *bk*. We applied Viola–Jones (Viola & Jones, 2004) face detection method on these images to extract the faces. Fig. 5 shows some samples from the database after face detection.

6.2. Accuracy

The accuracy of a biometric recognition system depends on different factors, *e.g.*, the discriminative ability of the features, number of features, and classifier performance. The previous methods discussed in Section 2 mainly depend on a special type of biometric data or recognition technique (Barni et al., 2010; Blanton & Gasti, 2011; Bringer et al., 2009, 2011, 2013a, 2013b; Erkin et al., 2009; Huang et al., 2011; Osadchy et al., 2010, 2013; Sadeghi et al., 2009). However, CloudID is not limited to any biometric trait, feature extraction technique or dimensionality reduction method, and it just deals with feature vectors. For example, the face recognition algorithm used in our experiments utilizes Gabor features and GDA dimensionality reduction. This combination with a simple k-nearest neighbor (KNN) classifier has better accuracy than the eigenface method (Turk & Pentland, 1991a) used by the previous privacy preserving solutions (Erkin et al., 2009; Sadeghi et al., 2009). Fig. 6 clearly shows that this approach outperforms the eigenface method.[4]

[2] Remember *SK* includes a private key for each possible predicate (query).

[3] It is noteworthy that for absolute data disclosure prevention, using a multiple-encryption scheme, the trusted party encrypts the records using a symmetric key such that M is already encrypted (ϵM) and the cloud provider is not allowed to see the actual records.

[4] Note that this comparison is between the face recognition approaches before applying any security scheme on them.

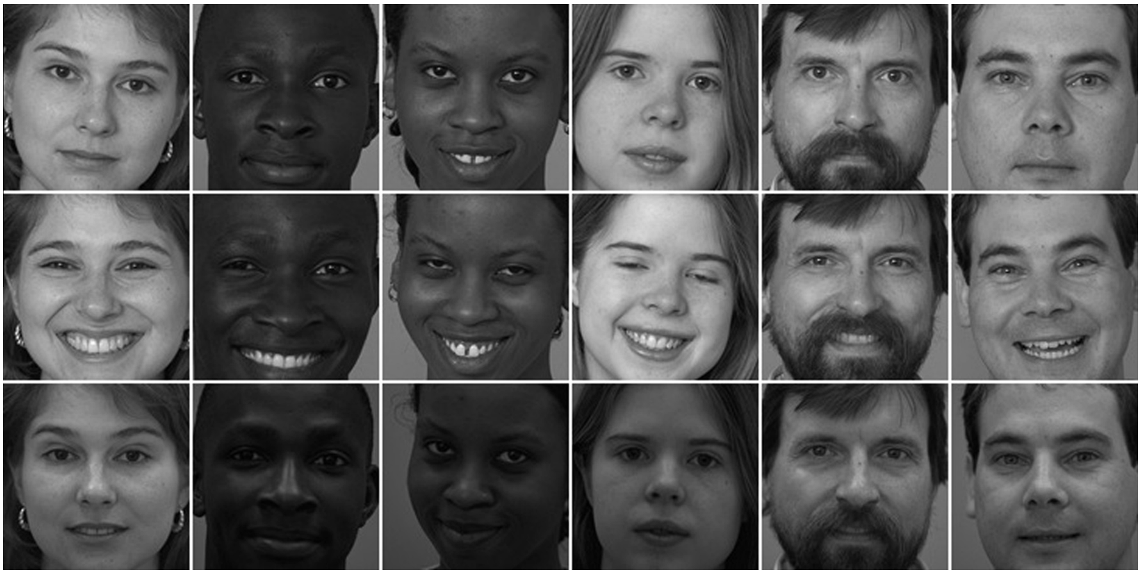

Fig. 5. Face images from FERET database after face detection and cropping. First row: frontal series (ba). Second row: frontal series with an alternative expression (bj). Third row: frontal series with illumination change (bk).

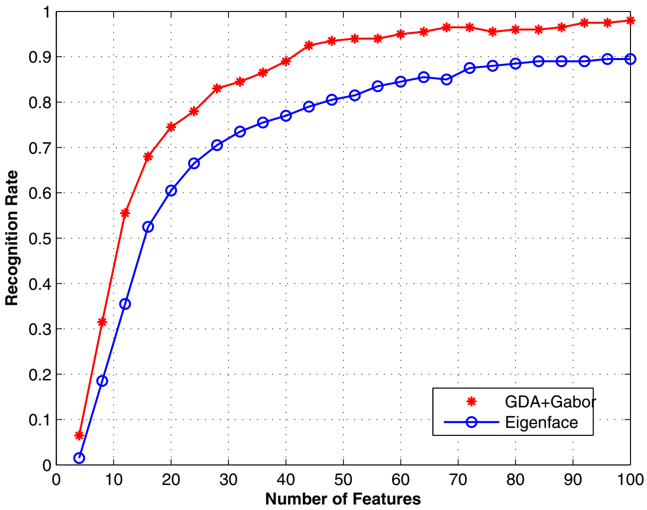

Fig. 6. Comparison of the eigenface and Gabor + GDA face recognition algorithms.

Table 1 shows the recognition and false positive rates of CloudID using different number of features. We chose the number of accepted adjacent intervals to be $2^{L-2}+2$. Each of the parallel trees may give different results for the identification problem. The matched subjects are the ones that majority of the trees have identified as the true matches. It is worth noticing that there is information loss in the process of quantizing the feature vectors. Therefore, the recognition accuracy also depends on the efficacy of the quantization method. The usage of quantization and range queries makes secure biometric systems feasible, but forces the designer to accept a reduction in accuracy in comparison to that which would have been achieved in an unconstrained scheme (Rane, Wang, Draper, & Ishwar, 2013).

Table 1
Face recognition accuracy of CloudID.

No. of trees	No. of features	Recognition rate (%)	False positive rate (%)
1	4	77.50	24.62
3	12	82.50	15.58
5	20	83.00	10.93
7	28	87.50	8.38
9	36	91.50	5.38
11	44	93.50	3.32
13	52	95.00	1.71

As shown in Table 1, increasing the number of features improves the recognition rate significantly. Moreover, since each proposition (matching feature) can be either *True* or *False*, if we hypothetically assume that each proposition has the same probability of being *True*, *e.g. p*, using *n* features in the conjunctive query, the probability of the conjunctive proposition being *True* is p^n. Hence, using more features decreases the probability of having a positive response exponentially which on the other hand leads to a reduction in false positive rate (see Table 1).

As mentioned before, we apply a range query over the encrypted database where the recognition rate is a function of the width of the accepted range. However, increasing the width of the range also increases the false positive rate. Table 2 clearly shows this effect for a fixed number of features ($n = 52$). Note that, the number of quantization levels increases exponentially relative to the level of the tree ($2^{L-1}+1$); therefore, the number of accepted adjacent intervals should also increase with the increase of the tree level to make sure that we cover almost the same size range covered in previous levels. Therefore, in addition to the number of features, the width of the accepted range is another degree of freedom for the designer.

Table 2
System accuracy using different accepted range sizes.

No. of accepted adjacent intervals	Recognition rate (%)	False positive rate (%)
$2^{L-2}-3$	28.50	0.00
$2^{L-2}-2$	44.00	0.01
$2^{L-2}-1$	59.50	0.07
2^{L-2}	70.00	0.26
$2^{L-2}+1$	84.00	0.95
$2^{L-2}+2$	95.00	1.71
$2^{L-2}+3$	98.00	6.39

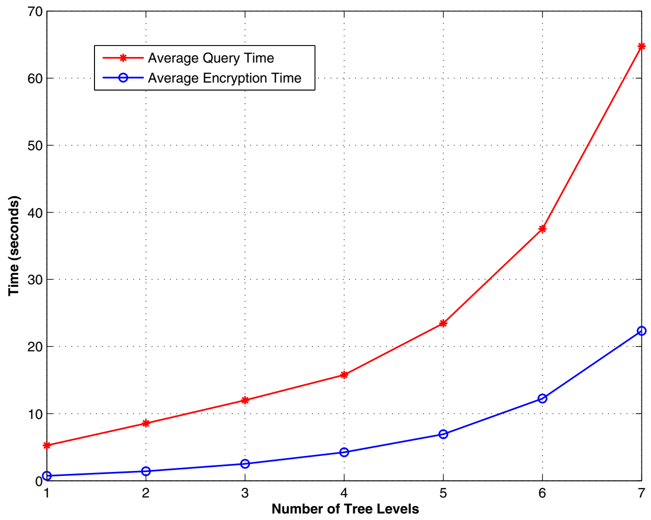

Fig. 7. Average encryption time and query time using different number of features.

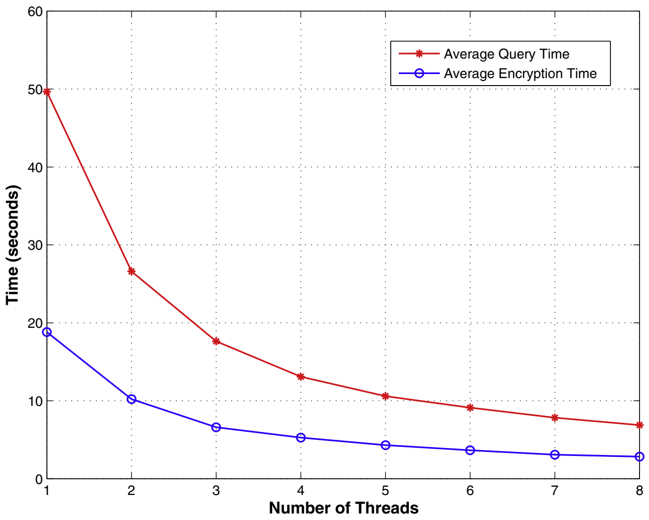

Fig. 8. Average encryption time and query time using multi-threading.

6.3. Performance

Quantizing the feature vectors and using range queries to have privacy-preserving properties forces a compromise between the discriminative power and the security level of the feature values. This, in turn, affects the accuracy of the system, highlighting the fact that privacy comes at the price of performance (Rane et al., 2013). For real-world deployment, CloudID needs to respond to queries efficiently. Fig. 7 shows the average encryption time needed for each subject along with the average time needed to perform a query using different number of levels in each tree. The experimental results clearly verify the exponential increase in complexity with the increase in the number of features in each tree, which is related to the number of predicates. On the other hand, the numbers of trees brings only a linear increase in complexity.

For a fixed number of predicates, the average encryption time for each subject is constant. In our experiments, using the last 4 levels of each tree, the average encryption time of each subject is about 18 s. However, as the number of subjects in the database increases, the number of comparisons for query processing grows as well, which increases the query time. Note that, in authentication problems, we would not have this complexity and the query could be made in real-time. But to avoid any information leakage, we do not label data stored in the cloud; therefore, we resort to the identification problem.

The main weakness of the proposed method is its computational complexity. To further improve CloudID's performance, we implemented the system in parallel threads on eight cores. Due to the independence of the comparisons in CloudID, parallel computation can be effectively applied. Fig. 8 shows the encryption and query times of the system using different numbers of threads. Since not all the processes are threaded, using all of the cores does not necessarily reduce the time by one eighth. It is obvious that in real cloud systems, more resources can be employed, which can further improve the performance of the system.

6.4. Case study

In this section an overall case study of the system is presented using the first subject in Fig. 5. Viola–Jones face detection (Viola & Jones, 2004) is applied to crop the face area of the image. Then, features are extracted using the Gabor wavelets shown in Fig. 3. Fig. 9 illustrates the face detection step along with the real parts of the results of applying Gabor filters to the face image. As can be seen in Fig. 9, Gabor filters extract the variations in different frequencies and orientations in the face. Here, the size of the resulting feature vector is the size of the image (120×120) multiplied by the number of scales and orientations (5×8) divided by the row and column downsampling factors (4×4) which is $120 \times 120 \times 5 \times 8/(4 \times 4) = 36{,}000$ in total.

After applying GDA dimensionality reduction, the number of features is reduced to $C - 1$, where C is 200 in our experiments. As mentioned in Section 4, the quantization is done using k-d trees. Each level of the k-d tree has 2^{L-1} boundary points which divide the space into $2^{L-1} + 1$ intervals. Therefore, each level will have $2^{L-1} + 1$ quantized values which are denoted by the integers 1 to $2^{L-1} + 1$. Assuming that we only use one tree with seven levels, below is the feature vector and its quantized form for the first image of the first subject, which is used for training and enrollment.

$$\mathbf{B}_{\mathbf{enroll}} = \begin{bmatrix} -4.9309 \\ -3.7765 \\ 5.1355 \\ 7.9747 \\ 5.7201 \\ 7.7592 \\ 1.719 \end{bmatrix} \xrightarrow{Quantization} \begin{bmatrix} 1 \\ 1 \\ 5 \\ 9 \\ 17 \\ 33 \\ 52 \end{bmatrix}$$

CloudID links the above vector to the records of the subject, encrypts them and stores them in the public cloud. During the time of query, let us assume that the image shown in the second row of the Fig. 5 is the image of the client's face. In this case, the feature vector for this sample is:

$$\mathbf{B}_{\mathbf{query}} = \begin{bmatrix} -4.1877 \\ -2.7772 \\ 3.6925 \\ 1.5925 \\ 1.8369 \\ 3.564 \\ 5.228 \end{bmatrix} \xrightarrow{Quantization} \begin{bmatrix} 1 \\ 1 \\ 5 \\ 9 \\ 15 \\ 30 \\ 59 \end{bmatrix}$$

The system feeds the query feature vector, B_{query}, into the k-d tree and extracts lower and upper boundaries to create the

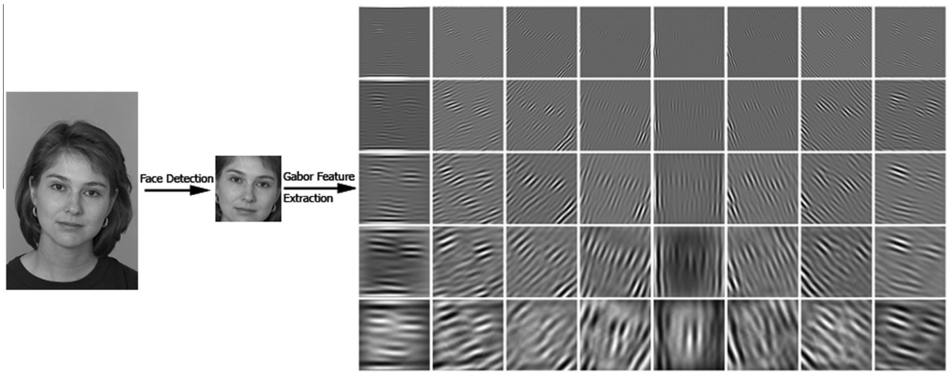

Fig. 9. Face detection and feature extraction, results of applying filters shown in Fig. 3 on a face image.

predicate. Here in our experiments, we chose the number of accepted adjacent intervals to be $2^{L-2}+2$ to make sure that they cover almost the same size range for each feature. Note that in the quantization, we have considered the extremes. *i.e.*, minus infinity and plus infinity, to be 0 and $2^{L-1}+2$. Therefore, the boundaries will not exceed these extreme values. For example, for level 1, plus infinity is mapped to 3, while for levels 2 and 3, it is mapped to 4 and 6, respectively. As you can see in (23), the first three levels cover almost the whole quantized space from minus infinity (0) to plus infinity ($2^{L-1}+2$), *i.e.*, very coarse quantization, which is not useful for identification. Therefore, the system uses only the last 4 levels of the tree to create the predicate:

$$\begin{bmatrix} 0<1<3 \\ 0<1<4 \\ 1<5<6 \\ 3<9<10 \\ 5<15<18 \\ 12<30<34 \\ 25<59<66 \end{bmatrix} \Rightarrow P_o() = \begin{bmatrix} 3<b_4<10 \\ 5<b_5<18 \\ 12<b_6<34 \\ 25<b_7<66 \end{bmatrix} \tag{23}$$

The above predicate is used to create the token, which is used to query the encrypted database in the cloud. Any ciphertext whose corresponding feature vector falls in the intervals of the predicate will return *True* result for the conjunctive comparison. However, even if one of the propositions is *False*, the whole conjunctive comparison will be *False*.

Eq. (24) shows the proposition of the enrolled feature vector, B_{enroll}, in the predicate created by the query feature vector, $P_o()$. Here the result of the conjunctive comparison is *True* for the query. However, there might be feature vectors of other subjects, which also satisfy the above predicate. The number of these incorrectly identified subjects affects the false positive rate of the system.

$$P_o(B_{enroll}) = \begin{bmatrix} 3<p_4=9<10 \\ 5<p_5=17<18 \\ 12<p_6=33<34 \\ 25<p_7=52<66 \end{bmatrix} = True \tag{24}$$

Similarly, the above steps are applied to the other trees using other features. The overall response for the conjunctive query will be *True* only if majority of the trees have *True* response.

7. Conclusions and future work

In this paper, we presented a privacy-preserving cloud-based and cross-enterprise biometric identification solution. Our proposed system is the first cloud-based biometric identification system with a proven zero data disclosure possibility. In this approach, all biometric information are encrypted and the identification process is performed in the encrypted domain without decrypting the biometric data.

Unlike other privacy-preserving biometric identification methods, our approach is not limited to any special type of biometric data and it can work with any biometric trait and feature extraction techniques. Moreover, it does not use distance-based matching, which is proven to have information leakage. However, it applies a conjunctive range query over encrypted gallery samples, which returns *true* response only if all the features of the gallery sample fall in certain ranges defined by predicates created using the query sample. This makes CloudID secure against *center search attack* in which the attacker can recover the biometric template even if it is stored encrypted. We proposed a k-d tree structure to quantize the biometric feature vectors and define the range predicates. This structure also allows the system to handle variations in the biometric data.

The proposed system enables clients to securely store their confidential information in the cloud and facilitates remote biometric-based identification by enterprises that are granted access to these confidential records. It provides a solution to the concerns about the security and confidentiality of personal information stored in the cloud through the use of biometrics, while guarding against identity theft. We implemented a working prototype of the CloudID and evaluated it using a face biometric database. Our experimental results show that CloudID can be used in practice for biometric identification with a proven zero data disclosure.

The main weakness of our proposed method is its complexity and the size of the ciphertext. In order to perform real-time identification in case of large databases, more computing resources need to be allocated by the cloud provider, which might be costly. Another issue is that the identification accuracy is dependent on the efficiency of the quantization approach. In the future, we plan to solve the complexity problem of the algorithm using more efficient searchable encryption techniques. Moreover, we plan to design more intelligent and possibly adaptive quantization methods to reduce the information loss in this step and consequently increase the recognition rate. We also plan to design an expert system that can integrate multiple sources of biometrics information, *e.g.*, face, ear, and fingerprint, to make a more reliable recognition.

Appendix A. Proof of the security of the system

We review the proof of the security of CloudID's searchable encryption scheme presented in Boneh and Waters (2007). Let's define a security game in which an adversary is given a number of tokens and is required to distinguish two encrypted messages. The ith experiment in the game proceeds as follows:

- **Setup** – The challenger generates the public and secret keys and PK is passed to the adversary.
  $PK \leftarrow (PK_1, PK_2, \ldots, PK_t)$
  $SK \leftarrow (SK_1, SK_2, \ldots, SK_t)$
- **Query Phase I** – The adversary adaptively requests for the tokens of the predicates $P_1, P_2, \ldots, P_{q'} \in \Phi$, and the challenger responds with the corresponding tokens.
  $TK_j \leftarrow GenToken(SK, P_j)$.
- **Challenge** – The adversary chooses two data-biometric pairs (M_0, B_0) and (M_1, B_1) subject to the following restrictions:
  – $P_j(B_0) = P_j(B_1)$ for all $j = 1, \ldots, q'$.
  – If $M_0 \neq M_1$, then $P_j(B_0) = P_j(B_1) = 0$ for all $j = 1, \ldots, q'$.
  In ith experiment, the challenger constructs the following ciphertexts:
  $C_j \leftarrow \begin{cases} Encrypt(PK_j, M_0) & \text{if } P_j(B_0) = 1 \quad \text{and } j \geqslant i \\ Encrypt(PK_j, M_1) & \text{if } P_j(B_1) = 1 \quad \text{and } j < i \\ Encrypt(PK_j, \perp) & \text{o/w}, \end{cases}$ and returns
  $C \leftarrow (C_1, C_2, \ldots, C_t)$.
- **Query Phase II** – The adversary can request more tokens for predicates $P_{q'+1}, \ldots, P_q \in \Phi$ as long as they adhere to the above restrictions.
- **Guess** – The challenger flips a coin $\beta \in \{0, 1\}$ and gives $C_* = Encrypt(PK_{B_\beta}, M_\beta)$ to the adversary, who returns a guess $\beta' \in \{0, 1\}$ of β. The advantage of adversary in attacking the system is defined as
  $Adv = |Pr(\beta = \beta') - \frac{1}{2}|$.

If Exp^i is the probability that the adversary guesses $\beta' = 1$ in experiment i, in a chain of $t + 1$ experiments, the adversary's advantage can be calculated by the differences in the outer experiments.

$$Adv = |Exp^1 - Exp^{t+1}| \leqslant \sum_{i=1}^{t} |Exp^i - Exp^{i+1}|.$$

Since the public key system is semantically secure, $|Exp^i - Exp^{i+1}|$ and consequently adversary's advantage are negligible, which make the Φ-searchable system secure.

References

Barni, M., Bianchi, T., Catalano, D., Di Raimondo, M., Donida Labati, R., Failla, P., Fiore, D., Lazzeretti, R., Piuri, V., Scotti, F., et al. (2010). Privacy-preserving fingercode authentication. In *Proceedings of the 12th ACM workshop on multimedia and security* (pp. 231–240).

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation, 12*, 2385–2404. http://dx.doi.org/10.1162/089976600300014980.

Belhumeur, P. N., Hespanha, J. a. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*, 711–720. http://dx.doi.org/10.1109/34.598228.

Blanton, M., & Gasti, P. (2011). Secure and efficient protocols for iris and fingerprint identification. In *Computer security–ESORICS* (pp. 190–209). Springer.

Boneh, D., & Waters, B. (2007). Conjunctive, subset, and range queries on encrypted data. In *Proceedings of the 4th conference on theory of cryptography* (pp. 535–554). Springer-Verlag.

Bringer, J., Chabanne, H., Favre, M., Patey, A., Schneider, T., & Zohner, M. (2014). GSHADE: Faster privacy-preserving distance computation and biometric identification. In *Proceedings of the 2nd ACM workshop on information hiding and multimedia security* (pp. 187–198). ACM.

Bringer, J., Chabanne, H., & Kindarji, B. (2011). Identification with encrypted biometric data. *Security and Communication Networks, 4*, 548–562.

Bringer, J., Chabanne, H., & Patey, A. (2013a). Practical identification with encrypted biometric data using oblivious ram. In *International conference on biometrics (ICB)* (pp. 1–8). IEEE.

Bringer, J., Chabanne, H., & Patey, A. (2013b). SHADE: Secure hamming distance computation from oblivious transfer. In *Financial cryptography and data security* (pp. 164–176). Springer.

Bringer, J., Chabanne, H., & Kindarji, B. (2009). Error-tolerant searchable encryption. In *IEEE international conference on communications (ICC)* (pp. 1–6).

Daugman, J. G. (1993). High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15*, 1148–1161. http://dx.doi.org/10.1109/34.244676.

Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I., & Toft, T. (2009). Privacy-preserving face recognition. In *Privacy enhancing technologies* (pp. 235–253). Springer.

Fernandes, D. A., Soares, L. F., Gomes, J. V., Freire, M. M., & Inácio, P. R. (2014). Security issues in cloud environments: a survey. *International Journal of Information Security, 13*, 113–170.

Haghighat, M. B. A., & Namjoo, E. (2011). Evaluating the informativity of features in dimensionality reduction methods. In *5th international conference on application of information and communication technologies (AICT)* (pp. 1–5). IEEE.

Haghighat, M., Zonouz, S., & Abdel-Mottaleb, M. (2013). Identification using encrypted biometrics. In *Computer analysis of images and patterns (CAIP)* (pp. 440–448). Springer.

Huang, Y., Malka, L., Evans, D., & Katz, J. (2011). Efficient privacy-preserving biometric identification. In *Network and distributed system security symposium* (pp. 1–14).

Jain, A., Hong, L., & Pankanti, S. (2000a). Biometric identification. *Communications of the ACM, 43*, 90–98. http://dx.doi.org/10.1145/328236.328110.

Jain, A., Prabhakar, S., Hong, L., & Pankanti, S. (2000b). Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing, 9*, 846–859.

Jain, A. K., Ross, A., & Pankanti, S. (2006). Biometrics: A tool for information security. *IEEE Transactions on Information Forensics and Security, 1*, 125–143.

Javelin strategy & research. (2014). Smartphones, tablets, and fraud: When apathy meets security. URL: <https://www.noknok.com/sites/default/files/whitepapers/smartphonestabletsandfraudnoknokfinal.pdf>. accessed: 2014-11-18.

Kamarainen, J. K., Kyrki, V., & Kalviainen, H. (2006). Invariance properties of gabor filter-based features-overview and applications. *IEEE Transactions on Image Processing, 15*, 1088–1099. http://dx.doi.org/10.1109/TIP.2005.864174.

Kandukuri, B. R., Paturi, V. R., & Rakshit, A. (2009). Cloud security issues. In *IEEE international conference on services computing (SCC)* (pp. 517–520).

Khashman, A. (2008). Intelligent face recognition. In *Intelligence and security informatics* (pp. 383–406). Springer.

Liu, C., & Wechsler, H. (2002). Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing, 11*, 467–476.

Meshgini, S., Aghagolzadeh, A., & Seyedarabi, H. (2013). Face recognition using gabor-based direct linear discriminant analysis and support vector machine. *Computers & Electrical Engineering, 39*, 727–745.

Moses, Y., Adini, Y., & Ullman, S. (1994). Face recognition: The problem of compensating for changes in illumination direction. In *European conference on computer vision (ECCV)* (pp. 286–296). Springer.

Osadchy, M., Pinkas, B., Jarrous, A., & Moskovich, B. (2010). SCiFI – a system for secure face identification. In *IEEE symposium on security and privacy (SP)* (pp. 239–254).

Osadchy, M., Pinkas, B., Jarrous, A., & Moskovich, B. (2013). System for secure face identification (SCIFI) and methods useful in conjunction therewith. US Patent 8,542,886.

Padilha, R., & Pedone, F. (2015). Confidentiality in the cloud. *IEEE Security & Privacy*, 57–60.

Pagnin, E., Dimitrakakis, C., Abidin, A., & Mitrokotsa, A. (2014). On the leakage of information in biometric authentication. In *Progress in cryptology–INDOCRYPT 2014* (pp. 265–280). Springer.

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology (EUROCRYPT99)* (pp. 223–238). Springer.

Phillips, P. J., Moon, H., Rizvi, S. A., & Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*, 1090–1104.

Rane, S., Wang, Y., Draper, S., & Ishwar, P. (2013). Secure biometrics: Concepts, authentication architectures, and challenges. *IEEE Signal Processing Magazine, 30*, 51–64.

Ren, K., Wang, C., Wang, Q., et al. (2012). Security challenges for the public cloud. *IEEE Internet Computing, 16*, 69–73.

Sadeghi, A., Schneider, T., & Wehrenberg, I. (2009). Efficient privacy-preserving face recognition. *Information, security and cryptology–ICISC 2009*, 229–244.

Shen, L., Bai, L., & Fairhurst, M. (2007). Gabor wavelets and general discriminant analysis for face identification and verification. *Image and Vision Computing, 25*, 553–563. http://dx.doi.org/10.1016/j.imavis.2006.05.002.

Song, D. X., Wagner, D., & Perrig, A. (2000). Practical techniques for searches on encrypted data. In *IEEE symposium on security and privacy* (pp. 44–55). http://dx.doi.org/10.1109/SECPRI.2000.848445.

Takabi, H., Joshi, J. B., & Ahn, G.-J. (2010). Security and privacy challenges in cloud computing environments. *IEEE Security & Privacy, 8*, 24–31.

Turk, M., & Pentland, A., (1991). Face recognition using eigenfaces. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 586–591).

Turk, M., & Pentland, A. (1991a). Eigenfaces for recognition. *Journal of Cognitive Neuroscience, 3*, 71–86.

University of California Corporate Compliance Policies and Procedures. 2008. Legal medical record standards, policy no. 9420. URL: <http://policy.ucop.edu/doc/1100168/LegalMedicalRecord>.

University of Miami. 2015. Pegasus HPC System. URL: <http://ccs.miami.edu/hpc/?page_id=4872>. accessed: 2015-04-18.

Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision, 57*, 137–154.